Ada Is It The Best Language For DoD?

In 1974, each Military Department independently proposed the adoption of a common programming language for use in development of its own major weapon systems. A study done by Dr David Fisher 1977, p7, states there are 450 general-purpose languages and dialects currently used in the DoD, the only languages used in data processing and scientific applications are, respectively, Cobol and Fortran. A larger number of programming languages are used in embedded computer systems applications. (Dr Fisher 1976,p.6)

The key issue is the difference between "version" and "dialect". The term dialect indicates a relatively minor change in a language whereas version indicates larger change and usually has a different name although the new name may be a concatenation of a different year or number to the baseline (e.g. Jovial, Jovial 73)

More than 150 million SLOC (i.e. 81%) of the weapon system software is written and maintained in third generation languages. Ada having 49.70 million SLOC and Jovial 73 12.68 Ada is the leading third generation language in terms of existing weapon system source lines of code. The goal of commonality within the weapon system community has not been reached yet even for military standards such as Jovial and CMS-2. Different versions of a language are almost always incompatible. Dialects of a version present subtle but not inconsequential porting problems, particularly when they are dialects based upon older versions of the language. For example, there are 10 or more different dialects

of pre-J73 JOVIAL still in use.

Ada 83 is being used in weapon system software that are being modernized. Using SLOC as a measure of usage, Ada ranks first in weapon systems. The fact that Ada usage is not greater in DoD could be due to several factors. First production quality Ada compilers and development tools were not available immediately after the language was adopted as a standard. There was a lag time of 4 to 5 years before compiler vendors could offer choices of Ada environments for high performance host/target machines. Second there is always inertia to overcome before change can occur and the resistance of the DoD software development community to DoD policy toward the use of Ada perpetuated that inertia. And third, it takes time to educate and train software engineers and managers to understand the language and to use it effectively.

Jovial is a legacy software being maintained by software support that modify code and/or provide data processing service. This application was developed by a contractor and is being maintained by the government using language versions and dialects chosen by the development contractor.

Even if only one language were used, software commonality, portability and interoperability would be imperfect. With modern programming languages and compilers, increased use of COTS products and re-use of software components, it is possible to produce applications with components written in different languages. Ada, with its specified pragma interfaces, is a language that is well suited to being used with other languages in multi-language applications.

The Marines successfully introduced Ada through the Portable Recording System (PRS). PRS is a tactical communications support software system. The entire project, including training, analysis, design, implementation and testing was completed in 5 1/2 months.

No difficult problems were encountered in interfacing the new Ada95 system with existing hardware and software.

The higher order language JOVIAL was developed for the U.S. Air Force by the Systems Development Corporation (SDC) in the late 1950's. The name originally recommended was OVIAL, for Our Own Version of the International Algebraic Language. Jules Swartz, a computer scientist then employed at SDC. At a January meeting it was suggested OVAIL be changed to JOVIAL for Jules Own Version of the International Algebraic language since Jules was conducting the meeting.
Program management Directive 9070, Oct 26 1983, directed activities necessary to Develop and maintain embedded systems.

Advantages of Ada 95

It is an embedded and real-time system and has a number of built in features such as modularity, information hiding, structuring tools for large inheritance and support of objected design methods. Ada is also superior to Jovial in terms of reliability. When judging the size of language, one must consider that the competent use requires much more than just that of the language; it requires comparable familiarity with the environment. As acknowledged by all who have ever seen Ada code, Ada programs

readability has a profound effect on program maintenance and was one of the successful guiding criteria in the design of the syntax of Ada. Program Management Directive 9070, Oct 26, 1983, directed activities necessary to develop and maintain embedded computer software standardization efforts within the Air force. The objective is to provide systems with improved performance, high reliability and lower lifecycle costs. Because Ada was originally commissioned by the U.S. Department of Defense and used Successfully for mission-critical military applications. Ada is also used internationally for a wide range of commercial software that demands the same level of reliability expected of military systems. Ada's reliability stems from the rigorous definition of the compiler language as well as from the requirement for a dependable run0-time environment. Ada program reliability begins at the design phase of a project and forms a kind of project level linguistic backplane throughout the entire software lifecycle. This ensures a high degree of consistency from design through deployment. Ada is intended to eliminate errors early in the development process as well as to provide a mechanism for managing errors during execution. The early detection and elimination of errors in an Ada program is a function of Ada's philosophy that a compiler should catch the maximum number of errors. An Ada compiler can support such high reliability because it is both type-safe and visibility-safe. Type safety extends to every part of the Ada programming process and ensures that little is left to guesswork. Althouogh trying to compile a program in Ada is sometimes frustrating for an Ada novice, the rational is that it is better to get it right the first time instead of continuously doing it over.

Ada also includes powerful mechanisms for handling run-time errors through source code. This feature is also designed for maximum safety and dependability. Run-time exceptions management in Ada is supported, in large part, by the type-safe feature of the language.

Nancy Leveson once said during a keynote speech at OOPSLA that Ada is a good language for the development of safe systems. Ada can eradicate a large portion of common programming errors at the outset. To achieve the objectives of standardization, computer software support tools had to be developed for Air Force standards:

- MIL-STD-1589, JOVIAL J-73 High Order Language (HOL)
- MIL-STD-1750A, Military Standard Sixteen-Bit Instruction Set Architecture

Advantages of JOVIAL

These support tools became the Integrated Tool Set (ITS). The roots of the ITS can be traced back to a JOVIAL compiler developed by the Air Force in the late 1970's. This was later joined by software tools for MIL-STD-1750A computers. These early tools were replaced when the Embedded Computer Standardization Program Office (ECSPO), ASD-AFALC/AXTS, Wright Patterson Air Force Base, Ohio, contracted for development of the ITS. Each ITS contains a JOVIAL compiler which can produce code executable on a VAX, PC, or MIL-STD-1750A assembler, linker and simulator/debugger. The ITS is required for on-board avionics, tactical and strategic missiles, munitions, and space systems. Maintaining these software support tools is an extremely complex task.

Optimization of compilers requires a great deal of knowledge concerning com instruction set architecture, plus keeping track of the cumulative effects of source language statements. Maintenance of the other tools requires similar knowledge and skills. Nineteen systems including the B52, F-15/16 and E-3 Sentry AWACS are weapon systems using JOVIAL. Failure to continue this effort will adversely impact current and future standardization efforts. The Air Force has invested approximately $16 million in development and maintenance of the ITS over more than 15 years. If the standard baseline configuration is not maintained, it will adversely affect DoD weapon system activities, i.e., legacy software reuse efforts, and Air Force development efforts that rely on the availability of support software for the JOVIAL HOL. Historically, a number of Air Force program offices have provided the ITS to software developing and supporting contractors as government-furnished property. Under such contracts, failure to update and correct software deficiencies could result in government default and allow the contractor to renegotiate weapon system costs and schedules under sole source conditions. For a specific weapon system, this could result in a significant cost increase to the Department of Defense. Overall software support costs would increase because separate funding would be required to maintain unique versions of ITS for each weapon system. Proliferation of the ITS would also increase, thus further reducing the commonality of weapon support software. The mission of the U.S. air Force JOVIAL ITS Program Office is to provide current and future customers with superior service and support distribution of the best JOVIAL compilers and MIL-STD-1750A tool sets office provides

a cost-effective way to maintain and modernize existing quality software products. In an effort to maintain JOVIAL development environments current with today's technology and provide a cost-effective way to modernize operational aircraft, the development of hosting JOVIAL compiler to RISC architecture is being pursued.

In conclusion; although Ada is an ideal language for an architecture-based environment, for certain sections of the Air Force, it is more cost effective to keep and maintain the current system.

REFERENCE LIST

Ada 95 Mapping Specification and Rationale, Version 4.1 Boston MA 1992

Institute for Defense Analysis Survey P-3054 Jan 95

MITRE (Avionics Domain)

Guidelines for Successful Acquisition and Management of Software Intensive Systems Vol I USAF Feb 1995

Lt General Carl G. O'Berry, Deputy Chief of Staff, Command, Control, Communications, and Computers, Headquarters US Air Force 1993

MIL-STD 1589, JOVIAL J-73 High Order Language (HOL)

MIL-STD 1750A, Military Standard Sixteen-Bit Instruction Set Architecture